Notebook

March 9, 2025

root/RAG_const/image_dataset_RAG_construction.ipynb

[1]:
```
from PIL import Image
import torch
import json
import numpy as np
import faiss
from transformers import CLIPProcessor, CLIPModel

#   CLIP
clip_model = CLIPModel.from_pretrained("/root/autodl-tmp/clip-vit-base-patch32")
clip_processor = CLIPProcessor.from_pretrained("/root/autodl-tmp/
 ↪clip-vit-base-patch32")

#
def get_image_embedding(image_path):
    image = Image.open(image_path).convert("RGB")
    inputs = clip_processor(images=image, return_tensors="pt", padding=True)
    with torch.no_grad():
        image_embedding = clip_model.get_image_features(**inputs)
    return image_embedding.numpy()

#
def get_text_embedding(text):
    inputs = clip_processor(text=text, return_tensors="pt", padding=True)
    with torch.no_grad():
        text_embedding = clip_model.get_text_features(**inputs)
    return text_embedding.numpy()

#
def load_image_text_data(json_file):
    with open(json_file, "r", encoding="utf-8") as f:
        data = json.load(f)
    image_texts = []
    for item in data:
        if item["type"] == "image_text":
            image_texts.append({
                "image_path": item["image"],
                "text": f"Q: {item['question']} A: {item['answer']}"
```

```
            })
    return image_texts

#     FAISS
def build_multimodal_faiss_index(image_texts):
    embeddings = []
    for item in image_texts:
        image_embedding = get_image_embedding(item["image_path"])
        text_embedding = get_text_embedding(item["text"])
        combined_embedding = np.concatenate([image_embedding, text_embedding],␣
 ↪axis=1)
        embeddings.append(combined_embedding)
    embeddings = np.vstack(embeddings)
    dimension = embeddings.shape[1]
    index = faiss.IndexFlatL2(dimension)
    index.add(embeddings.astype("float32"))
    return index, image_texts

#
def save_multimodal_retrieval_system(index, image_texts, index_file,␣
 ↪texts_file):
    faiss.write_index(index, index_file)
    with open(texts_file, "w", encoding="utf-8") as f:
        json.dump(image_texts, f, ensure_ascii=False, indent=4)

#
def build_and_save_multimodal_retrieval_system(json_file, index_file,␣
 ↪texts_file):
    image_texts = load_image_text_data(json_file)
    index, image_texts = build_multimodal_faiss_index(image_texts)
    save_multimodal_retrieval_system(index, image_texts, index_file, texts_file)
    print("        ")

#
image_text_data_file = "/path/to/your/image_text_data.json"
image_index_file = "/path/to/save/image_index.faiss"
image_texts_file = "/path/to/save/image_texts.json"
build_and_save_multimodal_retrieval_system(image_text_data_file,␣
 ↪image_index_file, image_texts_file)
```

```
---------------------------------------------------------------------------
TimeoutError                              Traceback (most recent call last)
File ~/miniconda3/lib/python3.12/site-packages/urllib3/connection.py:203, in␣
 ↪HTTPConnection._new_conn(self)
    202 try:
--> 203     sock = connection.create_connection(
    204         (self._dns_host, self.port),
```

```
    205         self.timeout,
    206         source_address=self.source_address,
    207         socket_options=self.socket_options,
    208     )
    209 except socket.gaierror as e:

File ~/miniconda3/lib/python3.12/site-packages/urllib3/util/connection.py:85, in␣
 ↪create_connection(address, timeout, source_address, socket_options)
     84 try:
---> 85     raise err
     86 finally:
     87     # Break explicitly a reference cycle

File ~/miniconda3/lib/python3.12/site-packages/urllib3/util/connection.py:73, in␣
 ↪create_connection(address, timeout, source_address, socket_options)
     72     sock.bind(source_address)
---> 73 sock.connect(sa)
     74 # Break explicitly a reference cycle

TimeoutError: timed out

The above exception was the direct cause of the following exception:

ConnectTimeoutError                       Traceback (most recent call last)
File ~/miniconda3/lib/python3.12/site-packages/urllib3/connectionpool.py:790, in␣
 ↪HTTPConnectionPool.urlopen(self, method, url, body, headers, retries,␣
 ↪redirect, assert_same_host, timeout, pool_timeout, release_conn, chunked,␣
 ↪body_pos, preload_content, decode_content, **response_kw)
    789 # Make the request on the HTTPConnection object
--> 790 response = self._make_request(
    791     conn,
    792     method,
    793     url,
    794     timeout=timeout_obj,
    795     body=body,
    796     headers=headers,
    797     chunked=chunked,
    798     retries=retries,
    799     response_conn=response_conn,
    800     preload_content=preload_content,
    801     decode_content=decode_content,
    802     **response_kw,
    803 )
    805 # Everything went great!

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connectionpool.py:491, in␣
 ↪HTTPConnectionPool._make_request(self, conn, method, url, body, headers,␣
 ↪retries, timeout, chunked, response_conn, preload_content, decode_content,␣
 ↪enforce_content_length)
```

```
    490         new_e = _wrap_proxy_error(new_e, conn.proxy.scheme)
--> 491     raise new_e
    493 # conn.request() calls http.client.*.request, not the method in
    494 # urllib3.request. It also calls makefile (recv) on the socket.

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connectionpool.py:467, i␣
 ↪HTTPConnectionPool._make_request(self, conn, method, url, body, headers,␣
 ↪retries, timeout, chunked, response_conn, preload_content, decode_content,␣
 ↪enforce_content_length)
    466 try:
--> 467     self._validate_conn(conn)
    468 except (SocketTimeout, BaseSSLError) as e:

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connectionpool.py:1096,␣
 ↪in HTTPSConnectionPool._validate_conn(self, conn)
   1095 if conn.is_closed:
-> 1096     conn.connect()
   1098 if not conn.is_verified:

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connection.py:611, in␣
 ↪HTTPSConnection.connect(self)
    610 sock: socket.socket | ssl.SSLSocket
--> 611 self.sock = sock = self._new_conn()
    612 server_hostname: str = self.host

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connection.py:212, in␣
 ↪HTTPConnection._new_conn(self)
    211 except SocketTimeout as e:
--> 212     raise ConnectTimeoutError(
    213         self,
    214         f"Connection to {self.host} timed out. (connect timeout={self.
 ↪timeout})",
    215     ) from e
    217 except OSError as e:

ConnectTimeoutError: (<urllib3.connection.HTTPSConnection object at␣
 ↪0x7fc8a0171850>, 'Connection to huggingface.co timed out. (connect␣
 ↪timeout=10)')

The above exception was the direct cause of the following exception:

MaxRetryError                             Traceback (most recent call last)
File ~/miniconda3/lib/python3.12/site-packages/requests/adapters.py:667, in␣
 ↪HTTPAdapter.send(self, request, stream, timeout, verify, cert, proxies)
    666 try:
--> 667     resp = conn.urlopen(
    668         method=request.method,
    669         url=url,
```

```
    670             body=request.body,
    671             headers=request.headers,
    672             redirect=False,
    673             assert_same_host=False,
    674             preload_content=False,
    675             decode_content=False,
    676             retries=self.max_retries,
    677             timeout=timeout,
    678             chunked=chunked,
    679         )
    681 except (ProtocolError, OSError) as err:

File ~/miniconda3/lib/python3.12/site-packages/urllib3/connectionpool.py:844, i␣
 ↪HTTPConnectionPool.urlopen(self, method, url, body, headers, retries,␣
 ↪redirect, assert_same_host, timeout, pool_timeout, release_conn, chunked,␣
 ↪body_pos, preload_content, decode_content, **response_kw)
    842     new_e = ProtocolError("Connection aborted.", new_e)
--> 844 retries = retries.increment(
    845     method, url, error=new_e, _pool=self, _stacktrace=sys.exc_info()[2]
    846 )
    847 retries.sleep()

File ~/miniconda3/lib/python3.12/site-packages/urllib3/util/retry.py:515, in␣
 ↪Retry.increment(self, method, url, response, error, _pool, _stacktrace)
    514     reason = error or ResponseError(cause)
--> 515     raise MaxRetryError(_pool, url, reason) from reason  # type:␣
 ↪ignore[arg-type]
    517 log.debug("Incremented Retry for (url='%s'): %r", url, new_retry)

MaxRetryError: HTTPSConnectionPool(host='huggingface.co', port=443): Max retrie␣
 ↪exceeded with url: /laion/CLIP-ViT-H-14-laion2B-s32B-b79K/resolve/main/
 ↪open_clip_pytorch_model.bin (Caused by ConnectTimeoutError(<urllib3.connectio␣
 ↪HTTPSConnection object at 0x7fc8a0171850>, 'Connection to huggingface.co time␣
 ↪out. (connect timeout=10)'))

During handling of the above exception, another exception occurred:

ConnectTimeout                            Traceback (most recent call last)
File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪1374, in _get_metadata_or_catch_error(repo_id, filename, repo_type, revision,␣
 ↪endpoint, proxies, etag_timeout, headers, token, local_files_only,␣
 ↪relative_filename, storage_folder)
   1373 try:
-> 1374     metadata = get_hf_file_metadata(
   1375␣
 ↪        url=url, proxies=proxies, timeout=etag_timeout, headers=headers, toke␣=token
   1376     )
   1377 except EntryNotFoundError as http_error:
```

```
File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/utils/_validator
 ↪py:114, in validate_hf_hub_args.<locals>._inner_fn(*args, **kwargs)
    112     kwargs = smoothly_deprecate_use_auth_token(fn_name=fn.__name__,␣
 ↪has_token=has_token, kwargs=kwargs)
--> 114 return fn(*args, **kwargs)

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪1294, in get_hf_file_metadata(url, token, proxies, timeout, library_name,␣
 ↪library_version, user_agent, headers)
   1293 # Retrieve metadata
-> 1294 r = _request_wrapper(
   1295     method="HEAD",
   1296     url=url,
   1297     headers=hf_headers,
   1298     allow_redirects=False,
   1299     follow_relative_redirects=True,
   1300     proxies=proxies,
   1301     timeout=timeout,
   1302 )
   1303 hf_raise_for_status(r)

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪278, in _request_wrapper(method, url, follow_relative_redirects, **params)
    277 if follow_relative_redirects:
--> 278     response = _request_wrapper(
    279         method=method,
    280         url=url,
    281         follow_relative_redirects=False,
    282         **params,
    283     )
    285     # If redirection, we redirect only relative paths.
    286     # This is useful in case of a renamed repository.

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪301, in _request_wrapper(method, url, follow_relative_redirects, **params)
    300 # Perform request and return if status_code is not in the retry list.
--> 301 response = get_session().request(method=method, url=url, **params)
    302 hf_raise_for_status(response)

File ~/miniconda3/lib/python3.12/site-packages/requests/sessions.py:589, in␣
 ↪Session.request(self, method, url, params, data, headers, cookies, files,␣
 ↪auth, timeout, allow_redirects, proxies, hooks, stream, verify, cert, json)
    588 send_kwargs.update(settings)
--> 589 resp = self.send(prep, **send_kwargs)
    591 return resp

File ~/miniconda3/lib/python3.12/site-packages/requests/sessions.py:703, in␣
 ↪Session.send(self, request, **kwargs)
```

```
    702 # Send the request
--> 703 r = adapter.send(request, **kwargs)
    705 # Total elapsed time of the request (approximately)

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/utils/_http.py:
 ↪93, in UniqueRequestIdAdapter.send(self, request, *args, **kwargs)
     92 try:
---> 93     return super().send(request, *args, **kwargs)
     94 except requests.RequestException as e:

File ~/miniconda3/lib/python3.12/site-packages/requests/adapters.py:688, in␣
 ↪HTTPAdapter.send(self, request, stream, timeout, verify, cert, proxies)
    687     if not isinstance(e.reason, NewConnectionError):
--> 688         raise ConnectTimeout(e, request=request)
    690 if isinstance(e.reason, ResponseError):

ConnectTimeout: (MaxRetryError("HTTPSConnectionPool(host='huggingface.co',␣
 ↪port=443): Max retries exceeded with url: /laion/
 ↪CLIP-ViT-H-14-laion2B-s32B-b79K/resolve/main/open_clip_pytorch_model.bin␣
 ↪(Caused by ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at␣
 ↪0x7fc8a0171850>, 'Connection to huggingface.co timed out. (connect␣
 ↪timeout=10)'))"), '(Request ID: b2cc3d7a-9e35-430e-80cf-b80d51b7879d)')

The above exception was the direct cause of the following exception:

LocalEntryNotFoundError                     Traceback (most recent call last)
File ~/miniconda3/lib/python3.12/site-packages/open_clip/pretrained.py:756, in␣
 ↪download_pretrained_from_hf(model_id, filename, revision, cache_dir)
    754 try:
    755     # Attempt to download the file
--> 756     cached_file = hf_hub_download(
    757         repo_id=model_id,
    758         filename=filename,
    759         revision=revision,
    760         cache_dir=cache_dir,
    761     )
    762     return cached_file  # Return the path to the downloaded file if␣
 ↪successful

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/utils/_validator
 ↪py:114, in validate_hf_hub_args.<locals>._inner_fn(*args, **kwargs)
    112     kwargs = smoothly_deprecate_use_auth_token(fn_name=fn.__name__,␣
 ↪has_token=has_token, kwargs=kwargs)
--> 114 return fn(*args, **kwargs)

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪860, in hf_hub_download(repo_id, filename, subfolder, repo_type, revision,␣
 ↪library_name, library_version, cache_dir, local_dir, user_agent,␣
 ↪force_download, proxies, etag_timeout, token, local_files_only, headers,␣
 ↪endpoint, resume_download, force_filename, local_dir_use_symlinks)
```

```
    859 else:
--> 860     return _hf_hub_download_to_cache_dir(
    861         # Destination
    862         cache_dir=cache_dir,
    863         # File info
    864         repo_id=repo_id,
    865         filename=filename,
    866         repo_type=repo_type,
    867         revision=revision,
    868         # HTTP info
    869         endpoint=endpoint,
    870         etag_timeout=etag_timeout,
    871         headers=hf_headers,
    872         proxies=proxies,
    873         token=token,
    874         # Additional options
    875         local_files_only=local_files_only,
    876         force_download=force_download,
    877     )

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪967, in _hf_hub_download_to_cache_dir(cache_dir, repo_id, filename, repo_type␣
 ↪revision, endpoint, etag_timeout, headers, proxies, token, local_files_only,␣
 ↪force_download)
    966     # Otherwise, raise appropriate error
--> 967     ␣
 ↪_raise_on_head_call_error(head_call_error, force_download, local_files_only)
    969 # From now on, etag, commit_hash, url and size are not None.

File ~/miniconda3/lib/python3.12/site-packages/huggingface_hub/file_download.py
 ↪1485, in _raise_on_head_call_error(head_call_error, force_download,␣
 ↪local_files_only)
   1483 else:
   1484     # Otherwise: most likely a connection issue or Hub downtime => let's␣
 ↪warn the user
-> 1485     raise LocalEntryNotFoundError(
   1486         "An error happened while trying to locate the file on the Hub␣
 ↪and we cannot find the requested files"
   1487         " in the local cache. Please check your connection and try agai␣
 ↪or make sure your Internet connection"
   1488         " is on."
   1489     ) from head_call_error

LocalEntryNotFoundError: An error happened while trying to locate the file on␣
 ↪the Hub and we cannot find the requested files in the local cache. Please␣
 ↪check your connection and try again or make sure your Internet connection is␣
 ↪on.

During handling of the above exception, another exception occurred:
```

```
FileNotFoundError                         Traceback (most recent call last)
Cell In[1], line 12
     10 CLIP_MODEL_NAME = "ViT-H-14"
     11 PRETRAINED_DATASET = "laion2b_s32b_b79k"  #
---> 12 clip_model, preprocess =␣
 ↪open_clip.create_model_and_transforms(CLIP_MODEL_NAME, pretrained=PRETRAINED_ )ATASET)
     13 tokenizer = open_clip.get_tokenizer(CLIP_MODEL_NAME)
     15 # **    **

File ~/miniconda3/lib/python3.12/site-packages/open_clip/factory.py:494, in␣
 ↪create_model_and_transforms(model_name, pretrained, precision, device, jit,␣
 ↪force_quick_gelu, force_custom_text, force_patch_dropout, force_image_size,␣
 ↪image_mean, image_std, image_interpolation, image_resize_mode, aug_cfg,␣
 ↪pretrained_image, pretrained_hf, cache_dir, output_dict, load_weights_only,␣
 ↪**model_kwargs)
    464 def create_model_and_transforms(
    465         model_name: str,
    466         pretrained: Optional[str] = None,
   (…)
    484         **model_kwargs,
    485 ):
    486     force_preprocess_cfg = merge_preprocess_kwargs(
    487         {},
    488         mean=image_mean,
   (…)
    491         resize_mode=image_resize_mode,
    492     )
--> 494     model = create_model(
    495         model_name,
    496         pretrained,
    497         precision=precision,
    498         device=device,
    499         jit=jit,
    500         force_quick_gelu=force_quick_gelu,
    501         force_custom_text=force_custom_text,
    502         force_patch_dropout=force_patch_dropout,
    503         force_image_size=force_image_size,
    504         force_preprocess_cfg=force_preprocess_cfg,
    505         pretrained_image=pretrained_image,
    506         pretrained_hf=pretrained_hf,
    507         cache_dir=cache_dir,
    508         output_dict=output_dict,
    509         load_weights_only=load_weights_only,
    510         **model_kwargs,
    511     )
    513     pp_cfg = PreprocessCfg(**model.visual.preprocess_cfg)
    515     preprocess_train = image_transform_v2(
    516         pp_cfg,
```

```
    517         is_train=True,
    518         aug_cfg=aug_cfg,
    519     )

File ~/miniconda3/lib/python3.12/site-packages/open_clip/factory.py:375, in␣
 ↪create_model(model_name, pretrained, precision, device, jit, force_quick_gelu␣
 ↪force_custom_text, force_patch_dropout, force_image_size,␣
 ↪force_preprocess_cfg, pretrained_image, pretrained_hf, cache_dir, output_dict␣
 ↪require_pretrained, load_weights_only, **model_kwargs)
    373 pretrained_cfg = get_pretrained_cfg(model_name, pretrained)
    374 if pretrained_cfg:
--> 375     checkpoint_path =␣
 ↪download_pretrained(pretrained_cfg, cache_dir=cache_dir)
    376     preprocess_cfg = merge_preprocess_dict(preprocess_cfg,␣
 ↪pretrained_cfg)
    377     pretrained_quick_gelu = pretrained_cfg.get('quick_gelu', False)

File ~/miniconda3/lib/python3.12/site-packages/open_clip/pretrained.py:794, in␣
 ↪download_pretrained(cfg, prefer_hf_hub, cache_dir)
    792         target = download_pretrained_from_hf(model_id,␣
 ↪filename=filename, cache_dir=cache_dir)
    793     else:
--> 794         target =␣
 ↪download_pretrained_from_hf(model_id, cache_dir=cache_dir)
    796 return target

File ~/miniconda3/lib/python3.12/site-packages/open_clip/pretrained.py:764, in␣
 ↪download_pretrained_from_hf(model_id, filename, revision, cache_dir)
    762     return cached_file  # Return the path to the downloaded file if␣
 ↪successful
    763 except Exception as e:
--> 764     raise FileNotFoundError(f"Failed to download file ({filename}) for␣
 ↪{model_id}. Last error: {e}")

FileNotFoundError: Failed to download file (open_clip_pytorch_model.bin) for␣
 ↪laion/CLIP-ViT-H-14-laion2B-s32B-b79K. Last error: An error happened while␣
 ↪trying to locate the file on the Hub and we cannot find the requested files i␣
 ↪the local cache. Please check your connection and try again or make sure your␣
 ↪Internet connection is on.
```

This notebook was converted with convert.ploomber.io